COMP4601Project Documentation

Introduction

Our project is a text summarization system which collects job ad web pages on some of the major job search websites and data collected from the Internet would be used as source for analyzing the employers' preference for certain computer science skills. The aim of our project is to help computer science graduates set up goals for learning new techniques, or simply to feed people's curiosity.

Design Pattern

The whole project is consist of three parts:

- WebCrawler
- COMP4601ProjectJobAdCollector
- COMP4601ProjectWebService

WebCrawler

is a simple self-customized web crawler program which is consist of two main classes under edu.carleton.webcrawler.main:

- UrlCollector
- WebPageCollector

UrlCollector

This class is to collect URLs with specified patterns in a given web page. The UrlCollectorOnFinishListener is called when the URL collection process is finished and a UrlSet is returned. UrlSet is simply a URL string container. setNumOfThreads method is used to set the number of thread involved in the

process of collecting web URLs.

WebPageCollector

This class is to collect web page content based on the given UrlSet. It uses Jsoup to parse level 1 to 4 of HTML titles and the paragraphs from the web page. The WebPageCollectorOnFinishListner is called when it finishes collecting web pages.

COMP4601ProjectJobAdCollector

Some of the important classes:

- JobAdCollector
- SeedConfig
- TextAnalyzer
- MongoDBManager

JobAdCollector is the implementation of the actual process of job ads collection. It calls the getSeed method in SeedConfig to get the seed URLs with specified matchers(patterns that the URL might contain). The WebPageCollectorOnVisitListener is called each time when a new page is visited and a Page object is returned. Page's content will be tokenized and the program will analyze each token using TextAnalyzer to check if a certain job skill keyword exits in the page content. If a skill keyword or its other forms is found, program will accumulate it to jobPost. After checking through the whole page content, the page is added to MongoDB using MongoDBManager.

Algorithm

Skill Keyword Accumulative Method

Once a skill keyword occurs in the page content, the skill keyword will be put to skillTags in JobPostDocument. The same skill keyword would not count once it is already in the skillTags.

Document Collection Accuracy

In order to gain higher accuracy. While collecting web pages, we only add the ones with titles that contains job title keywords to database. For example, if a job ad is found with title "Junior Web Programmer in Ottawa" would be consider as a Web Developer job.

Job Difficulty Ranking

The ranking is based on the number of skill keyword occurrence in the job ad. Because we think if there are many job skills are mentioned in a job ad, it would likely be a difficult job.